Applicant : Shackleford
Parent No. : n/a
Issued : n/a
Serial No. : 10/047,045
Filed : 1/14/2002
Page : 7

Attorney's Docket No.: 10017560-1

REMARKS

Examiner rejected claims 1-11 under 35 USC 112, second paragraph as being indefinite for failing to particularly point out and distinctly claim the subject matter which application regards as the invention. In particular, the Examiner indicated that the preamble of claim 1 was directed towards "searching a parent code sequence for a target code sequence" yet the Applicant did not specify anywhere in lines 3-13 of claim 1 a component to determine the target code sequence. Applicant respectfully submits that the matching circuit coupled to the shift register arrangement recited in claim 1 as filed clarifies the determination of the target code sequence component. Specifically, claim 1 recites in lines 7-10 "a matching circuit coupled to the shift register arrangement, the matching circuit adapted to ascertain code position matches between the subset of codes in the stages of the shift register arrangement and codes *in corresponding code positions of the target code sequence*, and provide a programmed binary value for each code position match" (emphasis added). This clearly indicates that the target code component is associated with the matching circuit as recited in claim 1. Accordingly, Applicant would request the Examiner withdraw this rejection of claim 1 and, by virtue of their dependence on claim 1, the corresponding rejection of claims 2-11.

In addition, the Examiner also rejected claim 1 under 35 USC 102b in view of Lipman et al. [Science 227:1425 (1985)]. To make a proper rejection under 35 USC 102b, Lipman must have every element of the claims with the same specificity as that claims. *See* Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 U.S.P.Q.2D (BNA) 1913, 1920 (Fed. Cir.), cert. denied, 493 U.S. 853, 107 L. Ed. 2d 112, 110 S. Ct. 154 (1989) (explaining that an invention is anticipated if every element of the claimed invention, including all claim limitations, is shown in

a single prior art reference) and see Jamesbury Corp. v. Litton Industrial Products, Inc., 756 F.2d 1556, 1560, 225 USPQ 253, 256 (Fed. Cir. 1985) (explaining that the identical invention must be shown in as complete detail as is contained in the patent claim).

Applicant respectfully submits that Lipman does not include any of the elements provided claim 1. Lipman does not provide "A circuit arrangement for searching a parent code sequence for a target code sequence" from the preamble of claim 1 but instead provides a general purpose microcomputer hardware with software program components. Claim 1 is clearly directed to a particular circuit and Lipman does not teach or even suggest implementing anything using custom circuits or hardware. With respect to hardware, Lipman merely compares performance of the Lipman algorithm on Personal Computers (i.e., the IBM PC– probably a 12 mHz 80286) and contrasts this performance with the higher performance Mini Computers (i.e., VAX 11/750) of the era (Abstract from Lipman).

Further, Lipman does not provide "a shift register arrangement having a plurality of stages, wherein each stage stores a code of a subset of codes of the parent code sequence, and the shift register arrangement is adapted to periodically shift the subset of codes to form a new subset of codes with another code from the parent code sequence in a leading stage" as recited in claim 1. Even if the general purpose microcomputer might contain one or more shift registers, Lipman does not specify using such shift registers in the manner described and provided for in claim 1. It is also not clear how, if at all, such registers would be harnessed from the teaching of Lipman.

In addition, Lipman does not provide "a matching circuit coupled to the shift register arrangement, the matching circuit adapted to ascertain code position matches between the subset of codes in the stages of the shift register arrangement and codes in corresponding code positions

of the target code sequence, and provide a programmed binary value for each code position match" as recited in claim 1. Instead, Lipman provides an amino acid replacement matrix using software (Abstract and pg. 1436, Col. 3, lines 21-24 of Lipman) to improve performance. Not only is Lipman not using a matching circuit but if Lipman were to have taught a circuit implementation it would result in a matrix and not a shift register arrangement.

Finally, Lipman does not provide "a pipelined adder arrangement coupled to the matching circuit, the adder arrangement adapted to sum the binary values for code position matches for each respective subset of codes" as recited in claim 1. Contrary to the Examiner's assertion, Lipman does not mention a pipeline operation hence would not be possible for Lipman to even work in a "pipeline fashion".

Clearly, Lipman cannot anticipate claim 1 or any other claim similar to claim 1 as more than one element is absent from the claim. *See* Kloster Speedsteel AB v. Crucible, Inc., 793 F.2d 1565, 1571, 230 U.S.P.Q. (BNA) 81, 84 (Fed. Cir. 1986) ("Absence from the reference of any claimed element negates anticipation.") Consequently, Lipman does not anticipate claim 1 as filed in the instant application. Although claims 2-11 are independently patentable over Lipman, they are also patentable for at least the same reasons as claim 1 due to their dependence on claim 1.

With respect to independent method claim 12, Lipman does not teach "shifting the parent code sequence through a shift register arrangement having a plurality of stages, wherein the shift register arrangement stores a subset of codes of the parent code sequence and each stage stores a code of the subset of codes, and each shift of the subset of codes forms a new subset of codes with another code from the parent code sequence in a leading stage" as recited in claim 12. Instead, Lipman locates and compares the beginning and end positions in sequences being

analyzed (pg. 1436, Col. 2, lines 13-17 of Lipman) rather than shifting a parent code sequence as recited in claim 12.

Further, Lipman does not teach "determining in parallel whether the codes in the stages of the shift register arrangement are equal to codes of the target code sequence in corresponding code positions, and generating in parallel signals of a programmed binary value for each equality of a subset code and a target code" also recited in claim 12. Lipman does not make mention of the word parallel or parallel processing throughout the article hence it would not be possible for Lipman to teach generating parallel signals as recited in claim 12.

In addition, Lipman does not teach "summing the signals of the programmed binary value in a pipelined adder that generates a sum corresponding to each shift of the shift register arrangement" as recited in claim 12. As previously mentioned, Lipman does not use pipelining or a shift register hence it would not be possible for Lipman to teach pipelined adding or operating a shift register. In fact, Lipman teaches selecting the beginning and end positions in the sequences (pg. 1436, Col. 2, lines 13-17 of Lipman) thus not operating on a sequence by shifting values but by selecting values from only the ends of any given sequence.

Because Lipman does not teach each element, it is also not possible for Lipman to anticipate claim 12 under 35 USC 102b. In addition to being independently patentable, dependent claims 13-22 include at least the same limitations as claim 12 and therefore also cannot be anticipated by claim 12.

Lipman also does not anticipate claim 23 under 35 USC 102b as it does not teach each and every element of this claim either. For example, Lipman does not teach "means for determining in parallel whether each code at a subset-code position in a subset of codes is equal to a code of the target code sequence in a corresponding target-code position, and generating in

parallel signals of a selected binary value for each equality of a subset code and the target code" as recited in claim 23. As previously mentioned, Lipman does not mention the word parallel or parallel processing in general hence it would not be possible to anticipate claim 23. For at least this reason alone, claim 23 is not anticipated and should be allowed in the instant application.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned "Version with markings to show changes made".

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Leland Wiesner, Applicants' Attorney at (650) 853-1113 so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Oct. 29, 2004

Date

Respectfully Submitted,

Leland Wiesner
Attorney/Agent for Applicant(s)
Reg. No. 39424

Leland Wiesner
Attorney
366 Cambridge Avenue
Palo Alto, California 94306
Tel. (650) 853-1113

Applicant : Shackleford
Parent No. : n/a
Issued : n/a
Serial No. : 10/047,045
Filed : 1/14/2002
Page : 7

Attorney's Docket No.: 10017560-1

Version with markings to show changes made

What is claimed is:

1. (Original) A circuit arrangement for searching a parent code sequence for a target code sequence, comprising:

a shift register arrangement having a plurality of stages, wherein each stage stores a code of a subset of codes of the parent code sequence, and the shift register arrangement is adapted to periodically shift the subset of codes to form a new subset of codes with another code from the parent code sequence in a leading stage;

a matching circuit coupled to the shift register arrangement, the matching circuit adapted to ascertain code position matches between the subset of codes in the stages of the shift register arrangement and codes in corresponding code positions of the target code sequence, and provide a programmed binary value for each code position match; and

a pipelined adder arrangement coupled to the matching circuit, the adder arrangement adapted to sum the binary values for code position matches for each respective subset of codes.

2. (Original) The circuit arrangement of claim 1, wherein each stage of the shift register arrangement is adapted for storage of a code of character data.

3. (Original) The circuit arrangement of claim 1, wherein each stage of the shift register arrangement is adapted for storage of a code of a plurality of character data.

4. (Original) The circuit arrangement of claim 1, wherein the pipelined adder arrangement is a pipelined adder tree.

5. (Original) The circuit arrangement of claim 1, wherein the pipelined adder arrangement includes at least one stage of pipelined carry-save adders coupled to at least one stage of pipelined carry-propagate adders.

6. (Original) The circuit arrangement of claim 5, wherein the at least one stage of pipelined carry-save adders are adapted to provide a plurality of binary vectors responsive to the quantity of code position matches, and the at least one stage of pipelined carry-propagate adders are adapted to add the plurality of binary vectors.

7. (Original) The circuit arrangement of claim 1, further comprising a pipelined summing circuit coupled to the pipelined adder arrangement and adapted to determine a moving sum of code position matches for a plurality of subsets of codes.

8. (Original) The circuit arrangement of claim 7, wherein the plurality of subsets of codes includes at least a most recent subset of codes and a next most recent subset of codes.

9. (Original) The circuit arrangement of claim 7, wherein the plurality of subsets of codes includes a first subset of codes and a prior subset of codes, wherein an intervening subset of codes is processed between the first subset of codes and the prior subset of codes.

10. (Original)The circuit arrangement of claim 1, wherein each subset of codes includes n contiguous codes from the parent code sequence.

11. (Original) The circuit arrangement of claim 1, wherein the matching circuit includes a plurality of programmable lookup tables, each lookup table having an input terminal coupled to an output terminal of a corresponding stage of the shift register arrangement and configured to provide a programmed value responsive to an input code value.

12. (Original) A method for searching a parent code sequence for a target code sequence, comprising:

shifting the parent code sequence through a shift register arrangement having a plurality of stages, wherein the shift register arrangement stores a subset of codes of the parent code sequence and each stage stores a code of the subset of codes, and each shift of the subset of codes forms a new subset of codes with another code from the parent code sequence in a leading stage;

determining in parallel whether the codes in the stages of the shift register arrangement are equal to codes of the target code sequence in corresponding code positions, and generating in parallel signals of a programmed binary value for each equality of a subset code and a target code; and

summing the signals of the programmed binary value in a pipelined adder that generates a sum corresponding to each shift of the shift register arrangement.

13. (Original) The method of claim 12, further comprising:

determining, for each respective subset of codes, a probability of being the target code sequence as the sum of the binary values for code position matches for the respective subset of codes divided a total quantity of code positions in the target code sequence; and associating the probability for each respective subset of codes with a unique identifier representative of a location within the parent code sequence at which the respective subset of codes exists.

14. (Original) The method of claim 12, wherein the parent code sequence represents a genome.

15. (Original)The method of claim 14, wherein each code of the parent code sequence is representative of a nucleotide type.

16. (Original)The method of claim 15, wherein the nucleotide type is selected from the group consisting of: adenine, thymine, guanine, and cytosine.

17. (Original) The method of claim 14, wherein the genome is a human genome.

18. (Original) The method of claim 12, further comprising:
configuring a plurality of lookup tables to generate respective signals of the programmed binary value when addressed by codes equal to codes of the target code sequence; and addressing the lookup tables with the codes of the subset of codes.

19. (Original)The method of claim 12, further comprising generating a moving sum, for n subsets of codes, of sums of the signals of the selected binary value.

20. (Original) The method of claim 19, wherein the n subsets of codes includes at least a most recent subset of codes and a next-most-recent subset of codes.

21. (Original) The method of claim 19, wherein the n subsets of codes includes a first subset of codes and a prior subset of codes, wherein an intervening subset of codes is processed between the first subset of codes and the prior subset of codes.

22. (Original) The method of claim 12, wherein each subset of codes includes m contiguous codes from the parent code sequence.

23. (Original) An apparatus for searching a parent code sequence for a target code sequence, each code in the parent code sequence having a parent-relative position, comprising:
means for periodically selecting subsets of codes of the parent code sequence, each code in the subset having a relative subset-code position defined by the parent-relative position, and each subset of codes differing from other subsets by parent-relative positions of the codes in the subset;
means for determining in parallel whether each code at a subset-code position in a subset of codes is equal to a code of the target code sequence in a corresponding target-code position, and generating in parallel signals of a selected binary value for each equality of a subset code and the target code; and
means for summing the signals of the selected binary value.